International Journal of Advanced Computer Research (ISSN (print): 2249-7277 ISSN (online): 2277-7970)
Volume-3 Number-4 Issue-13 December-2013

Framework for Efficient Secure Steganographic Communication over Network Protocols

Jasbir Singh[1], Lalitsen Sharma[2]

Abstract

Security by obscurity has gained attention; as a result steganography is becoming more popular in the network communication. Network steganography describes various methods used for transmitting data over a network without it being detected. Most of the methods proposed for hiding data in a network do not offer an additional protection to the covert data as it is sent as plain text. This paper presents a framework that offers the protection to the covert data by encrypting it and compresses it for gain in efficiency. Several ways of sending covert information through network using TCP/IP protocol are discussed. Also, the communication made more secure and efficient by using compression and encryption techniques. Finally, the performance of the proposed framework is compared with other steganography tools.

Keywords

Steganography, Covert Channel, Protocol Headers, Cryptography.

1. Introduction

The concept of a covert channel first came into existence due to Lampson [1]. Lampson described covert channel as a channel that is neither planned nor anticipated for communications, but is used for transmission of information. In other words, a covert channel transmits data obscurely using the bandwidth of some other channel. The covert channels are concerned with making the communications invisible in contrast to cryptography where the objective is to make the data impossible to read. From a network communication point of view, these covert channels can make use of network packets as the cover object. Large number of protocols in the Internet seems ideal to be vehicle for covert communication [2].

Jasbir Singh, Department of Computer Science & IT, University of Jammu, Jammu, India.
Lalitsen Sharma, Department of Computer Science & IT, University of Jammu, Jammu, India.

Although steganography is applicable to text, images, audio signals and other digital data, here network packet headers are chosen for this purpose.

Use of network protocols is usually done by making use of fields in protocol headers that are unused or redundant. To make the communication more secure, the covert data can be compressed and encrypted before it is hidden in the carrier.

In this paper the design of a framework for network steganography is proposed that offers additional security to the covert data by encrypting it. The implementation of the framework sends, receives and extracts encrypted messages hidden in TCP/IP protocol. Covert data is compressed for efficiency, and then encrypted before sending it. For encryption, RSA and AES-128 are used. The efficiency of the proposed implementation is then compared to other known steganography tools.

2. Literature Review

The term network steganography was first introduced in 2003 by Krzysztof Szczypiorski [3], referring to all the information hiding methods that may be use unused or reserved bits in packet headers, packet padding, or various header fields can be used as covert channels.

Data hiding in TCP/IP protocols has been illustrated earlier by Craig Rowland [4]. The author identified a number of fields within the protocol headers that are optional, reserved or kept unused during normal transmission. This paper extensively exploits the 16 bit IP packet identification field, the 32 bit TCP initial sequence number and acknowledgement sequence number fields for covert communication.

K. Ahsan et al. [5] presented a number of ways of practically hiding data inside TCP/IP protocol headers. The authors illustrate covert communication by packet sorting as well as by manipulating packet headers. In this paper, the Do Not Fragment (DF) bit in the Flags field of the IP header is manipulated for data hiding. The constraint on using DF bit for covert communication is that it requires prior knowledge of Maximum Transmission Unit.

J. Griffin et al. [6] presented a scheme for covert communication by manipulating the lower order bits of the TCP timestamp field. The lower order bits of their timestamps can be changed by enforcing small delays on the processing of chosen TCP packets. The lower order bit of the TCP timestamp are usually random in many network connections. Especially on a slow connection the lower order bits of TCP timestamps are randomly distributed since they are entirely based on the internal timings of the host. By changing the timing within the kernel, the value of the lower order bit can be selected. The values are chosen using a statistically random distribution, so that it is impossible to distinguish them from the unchanged values. The rewritten TCP timestamps must be monotonically increasing and reflect a reasonable progression of time.

In 2005, S. J. Murdoch et al. [7] analyzed the TCP/IP header fields that can be used for stuffing data in a covert manner. The conclusion is that most of the fields that have been used so far (like IP Type of Service and identification, or TCP initial sequence number and timestamp) can easily be detected by using a passive warden.

In 2010, B. Jankowski et al. [8] presented a Steganographic system which is the information hiding solution based on inter-protocol steganography. It may be deployed in LANs and it utilizes two protocols Ethernet and ARP/TCP to enable secret data exchange. A Steganogram is inserted into Ethernet frame padding but one must always "look" at the other layer protocol (ARP or TCP) to determine whether it contains secret data or not.

Digital steganography tools use different cover media for hiding data. For example Hide and Seek is one of the older methods of Steganography [9]. It uses a common approach and is relatively easy to apply in image and audio files. Steganography by this method is carried out by taking the low order bit of each pixel and using it to encode one bit of a character. It creates some noise in the image unless a grayscale image is used. The S-Tools package was written by Andy Brown [9]. Version 4 can process image or sound files using a single program. S-Tools involve changing the least significant bit of each of the three colors in a pixel in a 24-bit image. HIP was created by Davi Tassinari de Figueiredo in 2002. Hide In Picture (HIP) uses bitmap images. If the file to be hidden is large, it may be necessary to modify more than a single bit (LSB) from each byte of the image, which can make this difference more visible. HIP uses a pseudo-random number generator to choose the place to write each bit. The values given by the pseudo-random number generator depend on your password, so it is not possible for someone trying to read your secret data to get the hidden file without knowing the password. StegoMagic hides files or messages inside text, WAV and BMP files. The restriction of the software is that the file to be hidden can be a maximum of 1/8 of the size of the file it is hidden in. The software is operated through a simple GUI and also requires passwords. The text hiding uses tabs and new lines to encode each character.

3. Framework

The framework for efficient network steganography is presented in Figure 1. Two communicating parties denoted as Sender and Receiver, transfer information overtly over a computer network, and employ data hiding involving the TCP/IP protocol suite to communicate information covertly.

The sender hides a message in the protocol header and sends it to the receiver, so that anyone spying on the on the channel is unable to notice the message. The sender takes as input the covert message C_k. The message is first compressed as TCP/IP protocol headers are small and large quantities of data would require many packets. Due to which the communication could take time and a longer conversation means a longer period for a sniffer to notice the covert data.

After compression the message is then encrypted and steganographically embedded into packet headers of a sequence of network packets $\{P_k\}$ to generate a steganographic network packet sequence $\{S_k\}$. The steganographic network packet sequence $\{S_k\}$ is sent to the receiver over a computer network.

The receiver must be waiting to receive packets with covert data from a sender. The receiver first receives the packets and extracts the useful data from the headers. Then, the information is decrypted by the decryption algorithm and finally decompressed by the decompression algorithm to obtain the original message.

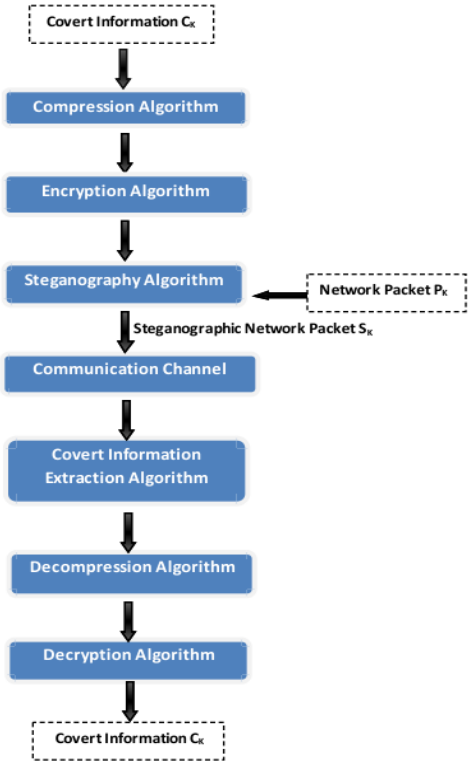

Figure 1: Covert channel framework for network steganography Adapted from K. Ahsan et al. [5]

4. Implementation

The framework is implemented in java using netbeans IDE based on WinPcap, an open source library for packet capture and network analysis. The sender determines the protocol and the file containing the message. The sender also specifies which encryption algorithm is to be used and whether compression is to be applied or skipped. The first two bytes of the message are used to indicate the encryption algorithm, application of compression and length of the message. The first bit indicates whether compression is to be applied (1) or skipped (0), second bit indicates the encryption algorithm used RSA (0) or AES-128 (1) and rest of the 14 bits specify the length of the message (0-16KB) in bytes. The message is compressed using the LZW [10] algorithm and written into a temporary file. LZW is a universal lossless data compression algorithm that takes advantage of this repetition. This method was originally introduced in 1978 by Lempel and Ziv and was later on modified in 1984 by Welch. For decompression also the same algorithm is used at the receiver end. The compression process can be skipped for small files. The compressed message is then encrypted either using either RSA [11] or AES-128 [12] algorithm. The message needs to split up into parts before being embedded in the packet headers. The encryption process might not be necessary in a secure environment. The selected protocol headers are then altered by stuffing the compressed and encrypted form of covert data into their chosen fields.

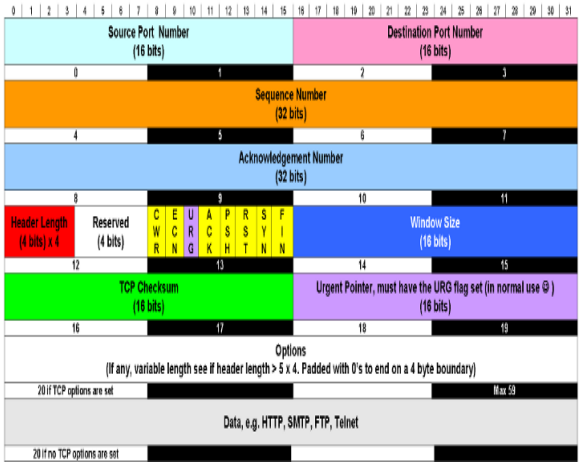

Figure 2: TCP header
Source: http://4.bp.blogspot.com/-xoglub4xhtk/tvinq0zuwzi/aaaaaaadxe/stwvjqq1qso/s400/tcp-header-colour.png

In order to hide data in network packets, we used the TCP/IP protocol. TCP headers are used for hiding data and the field in which data is hidden is the initial sequence number (ISN). The ISN field is opted here because it offers to transmit 32 bit covert data in just one packet. The encrypted message is divided into equally sized parts of four bytes. The ISN field of the TCP header is used for embedding these four byte parts.

The IP header provides only 16 bit for stuffing covert data. The field used for IP network steganography is the identification number (IP ID). According to [13], the IP ID is a value assigned by the sender which is used to distinguish fragments of a packet from fragments of another packet. Like ISN, the IP ID field is also required to be unpredictable and unique over a period of time.

The covert data is stuffed into the 16 bit identification field of the IP header using the method employed by Rowland [4]. This method requires the Do not Fragment (DF) bit in the Flags field of the IP header to be set on. This is done to make sure that the kernel does not alter the IP ID field during transmission of

packet. Also, in order to avoid fragmentation of packets, the size of all transmitted packets should be less than the Maximum Transmission Unit (MTU).

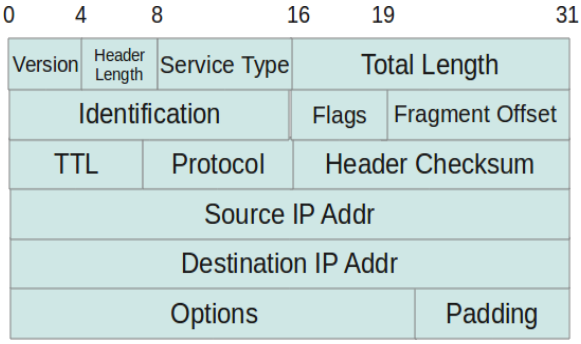

Figure 3: IP header
Source: http://static.thegeekstuff.com/wp-content/uploads/2012/03/ip-header-2.png

5. Results

The efficiency of proposed framework for network steganography is compared with other steganography tools. The text consisting of alphabets and numeric characters is generated randomly for performance testing and evaluation. This assists in attaining a healthy compression rate. A connection is established to communicate the size of the encrypted message. In every execution, six bytes of compressed and encrypted data is embedded inside the TCP and IP headers using a three way handshake. The message is split up into parts and stuffed into the TCP header by replacing the ISN with these parts. The message is split up into parts and stuffed into the TCP header by replacing the ISN and IP header by replacing the IP ID with these parts. Fields used for carrying covert data are listed in table 2.

Table 1: Efficiency comparison with other steganography tools

Tool	**Cover Media**	**Efficiency**
Hide and Seek	Audio file	0.3
S-Tools	Audio file	0.2
Hide In Picture (HIP)	Image file	0.15
StegoMagic	Text	0.125
Proposed Implementation	TCP/IP header	0.14

The efficiency is computed by dividing the size of the covert data transmitted with the size of the total amount of data transmitted. The size here is measured in bytes. This efficiency is compared with other steganography methods such as hiding data in audio files, hiding data in image files, hiding data in text files. To do so, we used the software applications presented at the last of section 2. All methods are tested with variable size inputs up to 1024 bytes and the observations are recorded. The mean of the observed values is obtained to represent the efiiciency. The results are shown in the table 1.

Table 2: Covert channels used

Protocol Header	**Field carrying the covert data**	**Offered bandwidth**
IP	IP Identification Field (IP ID)	2 bytes
TCP	Initial Sequence Number (ISN)	4 bytes

The proposed implementation shows results comparable to the method of hiding data in an image file.

6. Conclusion and Future Work

In this paper we presented a covert channel framework for network steganography using TCP/IP protocol. The framework provides additional security to the data sent through covert channel by using encryption. The implementation of the framework sends, receives and extracts encrypted messages hidden in TCP/IP protocol. Covert data is compressed for efficiency, and then encrypted using RSA and AES-128 before sending it. The efficiency of the proposed implementation is then compared to other known steganography tools and results obtained are comparable to the results of image steganography. Future work will continue in the direction of comparing Protocol Steganography with other forms of Steganography. Efforts will be to use more protocols for Protocol Steganography like the two less utilized protocols, ICMP and UDP and analyzing their performance in comparison to other Steganography methods.

References

[1] B. W. Lampson, "A note on the confinement problem", in Proc. of the Communications of the ACM, vol. 16, no. 10, pp. 613–615, Oct. 1973.
[2] S. Zander, G. Armitage and P. Branch, "A survey of covert channels and countermeasures in computer network protocols" IEEE communication surveys, 3rd Quarter 2007, vol. 9, no. 3, pp. 44-57.

[3] K. Szczypiorski. HICCUPS: Hidden Communication System for Corrupted Networks. In ACS '2003: Proceedings of The Tenth International Multi-Conference on Advanced Computer Systems, pages 31–40, Miedzyzdroje, Poland, 2003.
[4] C. Rowland. Covert Channels in the TCP/IP Protocol Suite. First Monday, Peer Reviewed Journal on the Internet, 2(5), January 1997.
[5] K. Ahsan and D. Kundur, "Practical Data Hiding in TCP/IP," in Proc. ACM Workshop. Multimedia Security, Dec. 2002.
[6] J. Giffin, R. Greenstadt, P. Litwack, and R. Tibbetts, "Covert Messaging Through TCP Timestamps." Massachusetts Institute of Technology - MIT, USA, 2002.
[7] S. J. Murdoch and S. Lewis. Embedding Covert Channels into TCP/IP. In Information Hiding: 7th International Workshop, volume 3727 of LNCS, pages 247–261. Springer, 2005.
[8] B. Jankowski, W. Mazurczyk, and K. Szczypiorski. Information Hiding Using Improper Frame Padding. CoRR, abs/1005.1925, 2010.
[9] P. Wayner, "Disappearing Cryptography". 2nd ed., Morgan Kaufmann Publishers, 2002.
[10] "Lempel–Ziv–Welch", online at http://en.wikipedia.org wiki/Lempe-Ziv-Welch (as of Dec., 2012).
[11] "RSA (algorithm)", online at http://en.wikipedia.org/wiki/RSA_(algorithm) (as of Jul., 2012).
[12] "Advanced Encryption Standard", online at http://en.wikipedia.org/wiki/Advanced_Encryption_Standard (as of Aug., 2012).

Jasbir Singh (Jammu, India, 30/10/2013) is MCA from University of Jammu, India and currently working as Assistant Professor in Department of Computer Science & IT in University of Jammu. His research interest is Network Security.

Lalitsen Sharma (Jammu, India, 30/10/2013) is MCA, Ph. D. for Guru Nanak Dev University, Amritsar, India and currently working as Associate Professor in Department of Computer Science & IT in University of Jammu. His research interest is Information Security.